12

EDGEWOOD

RESEARCH, DEVELOPMENT & ENGINEERING CENTER

U.S. ARMY CHEMICAL AND BIOLOGICAL DEFENSE COMMAND

ERDEC-TR-248

DESIGN AND IMPLEMENTATION OF A PROTOTYPE DATABASE SYSTEM FOR PROCESSING ENGINEERING DIRECTORATE INDIVIDUAL DEVELOPMENT PLANS

Allan C. Cleary

ENGINEERING DIRECTORATE

April 1995

Approved for public release; distribution is unlimited.

CUM SCIENTIA DEFENDIMUS

Aberdeen Proving Ground, MD 21010-5423

Disclaimer

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorizing documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No: 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1995 April	Final, 94 May - 94 Nov

4. TITLE AND SUBTITLE
Design and Implementation of a Prototype Database System for Processing Engineering Directorate Individual Development Plans

5. FUNDING NUMBERS
PR-1O162622A553C

6. AUTHOR(S)
Cleary, Allan C.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DIR, ERDEC, ATTN: SCBRD-ENC, APG, MD 21010-5423

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDEC-TR-248

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes the design and implementation of a prototype database system that was created to help the Engineering Directorate manage its Individual Development Plan (IDP) information. The database system was designed to meet the customers current needs and to be easily modifiable in the event that new needs are identified. The database uses a relational data model and the database manager includes operations for data retrieval and data management. Since mid October 1994, the database system has been used to process the directorates' FY95 IDP information.

14. SUBJECT TERMS
Database system
Individual development plan (IDP)

15. NUMBER OF PAGES
23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

BLANK

PREFACE

The work described in this report was authorized under Project No. 10162622A553C, Administrative and Management Support. This work was started in May 1994 and completed in November 1994.

The use of trade names or manufacturers' names in this report does not constitute an official endorsement of any commercial products. This report may not be cited for purposes of advertisement.

This report has been approved for release to the public. Registered users should request additional copies from the Defense Technical Information Center; unregistered users should direct such requests to the National Technical Information Service.

Acknowledgments

The author wishes to thank Frank Martin and Merlin Erickson for providing an environment that allows for learning and experimentation. This work could not have been completed without the dedication, support, and effort of George Pankoff, Fran Hughes, Barbara Knapp, Chris Jolley, Tiffany Wienecke, Nancy Chandler, Geri Cragg, Carol Hamilton, and Barbara Frank. Finally, thanks to Cecil Hassell, Doyle Stroup, Jeff Taylor, and Dick Dewey for taking the time to review this report.

BLANK

CONTENTS

1. INTRODUCTION 7

2. DESIGN 7

2.1 Requirements 8
2.2 Approach 9

3. IMPLEMENTATION 10

3.1 Database 10
3.2 Database Manager 12
3.2.1 Data Retrieval 12
3.2.2 Data Management 15

4. DISCUSSION 16

5. SUMMARY AND FUTURE WORK 16

5.1 Engineering Directorate Database Systems 16
5.2 Engineering Directorate IDP Process 17

LITERATURE CITED 19

APPENDIX A - IDP TOP LEVEL MENU 21

APPENDIX B - SAMPLE IDP FORM 23

BLANK

Design and Implementation of a Prototype Database System for Processing Engineering Directorate Individual Development Plans

1. INTRODUCTION

From 1992 through the end of this century, the skill base within the Engineering Directorate, Edgewood Research, Development and Engineering Center (ERDEC), will undergo changes. The traditional skill base will be eroded by personnel retirements and by transfers due to new technological opportunities. These new technological opportunities will be in non traditional areas, such as environmental technology, biotechnology and chemical demilitarization, and they will become increasingly important to the mission of the Chemical Biological Defense Command (CBDCOM).

Engineering Directorate management decided that one way to address the expected change would be to improve the existing Individual Development Plan (IDP) process. Now, it is Engineering Directorate policy to provide assistance to all employees in the development of their IDP; and, to try and support any IDP documented training and/or developmental assignment, where a mission need exists and funding is available. It must be noted that the IDP does not guarantee promotions; instead, it is a mechanism that the employee can use to prepare for the future in a manner consistent with the vision and values of the Command.

In FY 94, the employees of the Engineering Directorate were requested to develop and submit IDP's to the Directorate Professional Development Office. The employee response was favorable and approximately 300 IDP's were received. The downside was that the mechanism used to review and monitor the IDP's was tedious and labor intensive (the IDP's were sorted alphabetically, then placed into traditional manila file folders and accessed manually). After a thorough review of the FY 94 IDP process, directorate management decided to take steps to simplify the FY 95 process and to provide more efficient administrative support.

In May 94, the Applied Simulation Office, Engineering Directorate, was tasked to create a computer program to facilitate the management of the Directorate's FY 95 IDP information. Due to emerging customer requirements, the project allowed for a fair amount of experimentation and flexibility. Since the time frame was short, the project was divided into three distinct tasks (collect, format and process the data) to gain efficiency through a parallel approach. By mid October 1994, the FY 95 IDP's were collected, formatted and being processed. Since then, the program (process the data task) has been modified to respond to customer input and to optimize several of the internal data manipulation algorithms.

This report describes the design and implementation of a prototype database system that can be used to process and manage the Engineering Directorate's IDP information. The remainder of this report is organized as follows. The overall system requirements and design approach are presented in Section 2. An overview of the program, and some of interesting implementation aspects, is provided in Section 3. The manner in which the working database system is being used is discussed in Section 4; and, the paper is summarized, and potential future work is outlined, in Section 5.

2. DESIGN

While working on this project, it became increasingly apparent that the term "database" usually means different things to different people, at the same time. In this work, the following definitions (similar to definitions contained in [1]) will hold:

database - the information that is to be managed (i.e., it is the relevant data, and possibly some indexes, formatted and stored in some well defined way).

database manager - software that operates on the information that is stored in the database. In this work, the application (process and manage IDP data) and the database manager are contained within the same program.

database system - the database and the database manager.

2.1 Requirements

From the employee's standpoint, the IDP is a planning tool that can be used to create a "road map" for professional development and growth. From an organizational perspective, an IDP provides information about an employees current position and future goals that can be used in determining how the human resource allocation requirements for next generation and future systems/technologies will be met.

Early on, it was decided that the IDP would cover a five year period and that there would be no arbitrary restriction placed on the number of training opportunities that an employee could request. The primary customer, and owner, of the database system is the Engineering Directorate Professional Development Office. Other potential users include various components of the Engineering Directorate (e.g., the Department Heads, the Resource Control Office and the Administrative Department) . Based on discussions with the customer and the potential users, Table 1 was developed.

Requirement	Requester
Find IDP submissions that indicate changing career goals.	Prof. Dev. and Res. Control
Search all IDP submissions for common elements	Prof. Dev. and Admin.
View specified IDP submissions (i.e., IDP forms)	Prof. Dev. and Dept. Heads
Manage the database (add, delete, update information)	Prof. Dev.
Run on the CBDCOM pyramid computers	Prof. Dev. and Dept. Heads

Table 1. Database system requirements as defined by the users.

To meet the customer requirements, the design of the prototype database system was guided by the following set of technical goals.

(a) Be functional.

(b) Provide efficient information retrieval.

(c) Be easily modifiable and extendible.

(d) Be usable by personnel with varying computer skills.

Technical goal (b) deserves further comment. The primary use scenario (i.e., the only scenario that all of the customers agreed on), involves someone sitting at the computer and querying the database for information on personnel and their training needs; or searching the database for relationships (e.g., who has indicated a desire to attend the course entitled Geometric Dimensioning and Tolerancing and is also a GS-12?). It is anticipated that the need to manage the database will arise sporadically throughout a given year (i.e., the IDP information will remain fairly stable over one year periods).

2.2 Approach

Each IDP is viewed as a grouping of textual strings that represent personnel information about the individual who submitted it and the training events that they desire. In this context: personnel information contains data such as the persons name, grade, job series, current/future job specialty (e.g., test engineer, project manager), etc.; while, training events include courses (government or academic) and development assignments (Professional Long Term Training, details, etc.).

The personnel information can be structured easily, because it consists of atomic data elements (i.e., in terms of usable information each data element is fully decomposed, such as last name, grade or job series) and because it is contained in every IDP. Structuring of the training events, on the other hand, is less straightforward. Both the number and the types of training events may vary from one individual to the next (i.e., one can request x number of courses and y number of development assignments; and, theoretically, over a five year period, x ranges from 0 to about 25 and y ranges from 0 to about 5). In addition, each training event must be further decomposed to produce atomic data elements (e.g., data relevant to a training course includes the title of the course, the name of the institution that is offering the course, and the projected start date). It should be observed that allowing an IDP to contain an unrestricted number of training events rules out the use of a simple "flat file" database design (i.e., it would be extremely inefficient with respect to storage space and/or execution time).

The advantage of viewing each IDP as being composed of personnel information and training events is that it leads directly to a simple relational data model. Because an individual may request zero, one, or more than one training event, the personnel information and associated training events form a natural one-to-many relationship. Abstractly, a relational database is composed of *relations* that contain *tuples* (and each tuple has the same set of *attributes*); and, the relational database manager performs operations (e.g., scan, update) on the relations. In this report, the following terminology is used interchangeably: files or tables (*relations*), records or rows of the table (*tuples*), and fields or columns of the table (*attributes*).

Figure 1 shows a simplified example of a one to-many relationship between personnel information and training events, implemented as two tables. In Figure 1, the ID field forms the relation between the tables. Note that no semantic value is implied by the ID field (i.e., other than being unique, the ID field is not required to have a meaning nor does it imply that the data is ordered in any way.).

ID	L. Name	F. Name	Other
.	.	.	.
.	.	.	.
.	.	.	.
67	Street	Della	.....
.	.	.	.
.	.	.	.
.	.	.	.

Personnel Data

ID	Event	Course	Source	Time
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
67	Course	Calculus	JHU	1Q95
67	Course	Teaming	CBDCOM	2Q95
67	D. Assign	Detail	USACMLS	4Q95
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Training Data

Figure 1. Illustration of the one-to-many relationship between an employees personnel data and their training requests. Note that the ID field must be a unique identifier.

Conceptually, the prototype database system design involves partitioning the data into two or more separate tables (a personnel information table along with one or more training event tables); developing a mechanism for joining related information; and, providing the user with a set of data retrieval and maintenance operations. Note that the prototype database manager is designed to exploit the relational data model; but, it does not strictly adhere to the tenets of a relational database manager (the interested reader is referred to [2] for further information on relational database managers).

3. IMPLEMENTATION

3.1 Database

Since there are slightly less than 620 employees in the Engineering Directorate, the database will be composed of the information in, at most, 620 distinct IDP's. Based on the FY 94 IDP returns, the expected number of received IDP's was around 300. In any case, the amount of data is bounded and manageable. The data is partitioned into two files; one containing personnel information, and, the other containing the training events. Each file contains a well defined record and field structure.

The one-to-many relationship between the personnel information and the training event tables is realized by using an individuals CBDCOM computer user id. The user id was considered attractive for several reasons; almost 99% of the Engineering Directorate population has one, it is well known, understood and useful information, and best of all, each user id is guaranteed to be unique (plus they are managed by someone else). Two possible objections may be raised. The first involves the fact that the user id is a character string; and, the second involves the fact that 1% of the target population doesn't have one. The first will be countered below, where the implementation of the join is discussed (essentially, a function is applied to the user id that transforms it into an index into a a table). The second is countered by accepting the minimal risk that an individual without a user id may submit an IDP (and, by being willing to deal with it by creating a unique id when it happens).

The format of the data contained in the personnel information file is shown in Figure 2, part (a). Field 1 was discussed in the preceding paragraph; and, fields 2-5 are self explanatory. Fields 6-11 are implemented with numeric codes and they are three pairs ({6,7}, {8,9}, and {10,11}) of related information that allow the employee to indicate (or, the organization to recognize) career goals. For example, if an individual is currently a Quality Engineer who hopes to be a Project Manager someday, then they would enter the appropriate numeric codes into fields 10 and 11 to signal that. Finally, fields 12 and 13 are self explanatory; and, field 14 is used to facilitate data retrieval.

The detailed format of the data contained in the training event file is illustrated in Figure 2, part (b). A single record structure with six fields is used to store the data appropriate for a training event (i.e., a training course or a development assignment). In the case of a training course: field 2 indicates that it is a training course (and whether it scheduled in the current or in the out years); field 3 contains the course title; field 4 indicates if it is a government or non government facility; field 5 contains the institution offering the course; and, field 6 contains the projected start date. In the case of a development assignment: field 2 indicates that it is a development assignment; field 3 indicates the type (long term training or on the job training); field 4 indicates whether it is a government or non government facility; field 5 contains the location of the assignment; and, field 6 contains the projected start date.

It should be noted that the training event fields are more complex than those in the personnel information file and that the complexity can be easily eliminated by storing the training events in three separate files (e.g., current year courses, out year courses and development assignments). For the amount of data involved, it appears to be more efficient (in terms of execution time) to write code that handles the complexity than to accept the inherent cost of increased file activity (i.e., opening and closing the files).

(A) <u>Personnel Information</u>.

ID	LN	FN	MI	GR	PJ	FJ	PA	FJ	PS	FS	DE	TL	DA

where:

No.	Field	Contents
1	ID	CBDCOM computer user id
2	LN	Last name
3	FN	First name
4	MI	Middle initial
5	GR	Grade
6	PJ	Present job series
7	FJ	Future job series
8	PA	Present ACTEDS program
9	FA	Future ACTEDS program
10	PS	Present skill designator
11	FS	Future skill designator
12	DE	Department of record (i.e., Engineering, Support or Admin.)
13	TL	CBDCOM computer user id of team leader
14	DA	Development assignment (Yes or No)

(B) <u>Training Events</u>.

ID	FLAG	WHAT	WHERE	WHEN	TIME

where:

No.	Field	Contents
1	ID	CBDCOM computer user id
2	FLAG	DA - development assignment FU - training course scheduled for the out years PR - training course scheduled for the current fiscal year
3	WHAT	if FLAG = FU or PR, then title of the training course otherwise: PLTT - Professional Long Term Training program or, OJT - on the job training (e.g., a 120 day detail)
4	G/N	Government/non government facility
5	WHERE	if FLAG = FU or PR, then the Institution offering the course otherwise, the location of development assignment
6	WHEN	If FLAG = PR, then date contains the quarter (e.g., 2Q95) otherwise, the date is in fiscal year (e.g., FY97)

Figure 2. Format of the information contained in the personnel and training event data files.

3.2 Database Manager

The data base management program was written in the C programming language [3]; and, it executes on the CBDCOM computers. A printout of the top level menu is shown in Appendix A. Conceptually, the top level menu allows the user to retrieve information from the database, maintain the database, or exit the program. Currently, anyone may retrieve information, but only a small subset of users is permitted to perform the management operations (i.e., adding, deleting and updating data). The user interface is simple, character based, and menu driven; and, all inputs are case insensitive. A description of the data retrieval and the data maintenance aspects of the program, follows.

3.2.1 Data Retrieval

The top level menu provides three data retrieval options: find all of the IDP's in which differences exist between present and future categories (the field pairs, in fields 6-11, as discussed above); search the database, or view a persons entire IDP submission (i.e., form). A brief discussion of each option follows.

Selection of the find differences option, produces a second level menu that includes options for performing a specific find differences operation, for returning to the previous menu or for exiting the program. Three find differences options are currently supported: ACTEDS program, job series and skill specialty. Once an option is selected, the program generates a listing by identifying the rows in the personnel information table that contain different present and future information for the specified option. For example, if a user selects ACTEDS, the program searches for records where the present ACTEDS field does not match the future ACTEDS field. The program sorts the results by last name and outputs them in tabular form to the screen. The output includes each persons last name, first name, middle initial, present ACTEDS program and future ACTEDS program. After viewing the table, the user is given the option of saving it to a file.

According to the projected users, the database needed to be searched for specific things (either a discrete field or a Boolean "and" operation between two discrete fields). Therefore, providing a set of standard search options met the users needs and allowed for the development of fast, field specific, data searches. A subtle advantage of searching specific fields is that the output contains what the user expects it to contain (i.e., information relevant to the query). For example, if one used the partial pattern 11 with the grade search option, the result will contain all GS and WG 11's; while, a less specific search strategy would output all GS and WG 11's (relevant) plus all of the personnel information records with an 11 in any other field (irrelevant).

Selection of the search option produces a second level menu that includes options for returning to the previous menu, for accessing the numeric field code mappings, or for exiting the program. The following search options are currently supported.

1	ACTEDS program	6	Institution offering the course
2	Department of record	7	Gov. (or non Gov.) sponsored
3	Grade	8	Course title
4	Job series	9	Development assignments
5	Job specialty		

Options 1 through 5, refer to data that is contained in the personnel information table; while, options 6 through 9, refer to information that is contained in the training event tables.

Since, options 1-5 search for all of the field specific matches within the database, the user is requested to enter a single search pattern (e.g., if grade were selected, relevant patterns include GS11, or GS, or simply 11). Since options 6-9 perform a Boolean "and" operation on the specified fields, the user is asked to input two patterns (the desired pattern and the year). Note that with options 6-9, the user can retrieve all of the items by inputting a carriage return in response to the year prompt. In all cases, the results of a search are sorted by last name and

output in tabular form to the screen. After viewing the output table, the user is given the option of saving it to a file.

Data from the personnel information table is included in the output for all of the search options. For example, a grade search with the input GS11 results in a listing of the names (last, first, middle initial) of all personnel who submitted an IDP with GS11 in the grade field; a course title search with input of a course title and year results in a listing that includes the name of all personnel who submitted an IDP meeting that criterion. Extracting personnel information for use with the five personnel searches is straightforward, because the data required for the output resides in the same row of the table that is being searched. Retrieving personnel information for use with the training event searches requires an additional step; since the search is conducted within the training event table, a join operation is required to produce the appropriate personnel information.

The join operation is implemented with a hash and scan approach. While the program is executing, a table containing the personnel information and a hash table are both maintained in memory at all times (other tables, usually for training event information, are created and deleted within memory in response to the operations performed by a user). The personnel information table is implemented as an array of pointers, and each pointer in the table points to exactly one entity. Currently, each entity contains all of the information stored in one record of the personnel information file, plus a status field that is used to mark rows prior to deletion. Therefore, if the index into the table is known, the personnel information contained in any IDP within the database can be accessed directly.

The hash table provides a fast and effective way to lookup an index into the personnel information table. The hash table is implemented as an array of pointers, each pointer in the table points to the head of a linked list that contains elements that have the same hash value (calculated by a function that converts the persons user id into a non-negative integer). Each element in the hash table contains a user id and the index of the unique row that contains that user id in the id column of the personnel information table.

An illustration of a similar hash table with 10 rows is shown in Figure 3. Two things are worth observing: first, not all rows of the hash table contain elements (i.e., rows 3 and 8); and second, rows 0, 4 and 9, contain more than one element (also known as collisions). Theoretically, in the worst case, finding the index could take as many steps as there are rows in the personnel information table (e.g., if the hash function hashed all of the keys into the same row of the hash table). But, one can tune the hash function to spread the elements throughout the table, thereby reducing the number of collisions. Therefore, on average, the number of steps required to find the index will be bounded by a small constant (i.e., $O(1)$ in algorithmic terms). For further details on the O notation and hash tables, the interested reader is referred to [4].

Upon selecting the view option, the user is placed into a second level menu, where they are requested to input the last name of the desired individual. If the entered last name is unique to the database, the appropriate IDP form is generated and printed directly to the screen. If the specified last name is not unique to the database (e.g., currently there are six people with the last name Smith in the directorate), then a list containing the last name, first name and middle initial of all records containing the common last name is printed to the screen. Once the user selects the desired name from the list, the form is generated and printed to the screen. After viewing the form, the user is given the option of saving it to a file. A sample IDP form, and how it might be analyzed, is provided in appendix B.

Note that the view operation is tricky. A join operation is required to connect the appropriate training events with the personnel information; and, the user expects the training events to be ordered by date (note that the records in the training event table are not ordered by date). Currently, the join operation is accomplished in two steps. The first step involves finding all of the rows in the training event table whose id column contains the appropriate user id. The second step involves using the flag column to create three training event tables (current course, future course, development assignment). Then, prior to outputting the form, each of the three tables is sorted by its date field (the sorting is inexpensive since the size of each table is small, i.e., 2-5 entries).

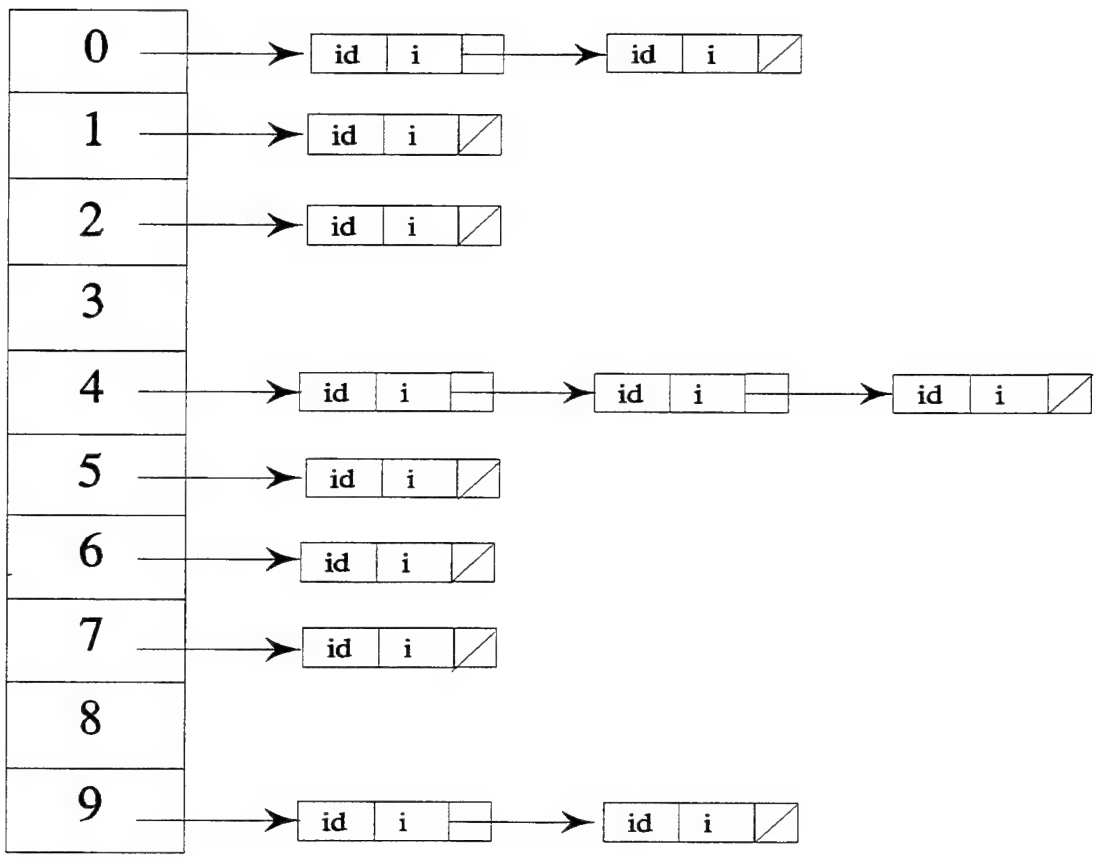

Figure 3. Illustration of a hash table. Slots 3 and 8 are empty, and slots 0, 4 and 9 contain collisions.

3.2.2 Data Management

Over time, the information in the database will need to be altered to reflect personnel actions and to reflect changes within the IDP's. For example a new IDP may be added when a person transfers into the directorate, an existing IDP may be deleted when an employee retires, or an employee may wish to update their current IDP to incorporate changing career goals. As a result, operations to manage the database are required. Since these operations involve writing the database, access levels and concurrency issues (e.g., simultaneous writes to the database that lead to data corruption) become relevant.

Both, access levels and concurrency issues, are mitigated by the expectation that less than five people will require access to the data management operations. Currently, the data management operations may only be accessed by personnel with "permission". Access permissions are implemented using the UNIX group facility. If the user is in the group, they can perform the operation; otherwise, a message indicating that the user is not authorized to perform the operation is printed to the screen. Based on the small number of users who have access to the management operations, a "do nothing" approach is taken with respect to concurrency (note that a simple mutual exclusion policy is being considered for a future version).

The top level menu provides three data management options: edit an existing IDP, delete an existing IDP, or add a new IDP. A discussion of each option follows.

Upon selection of the edit operation, the user is requested to input either the last name of the desired individual, or 'p' to return to the previous menu. The edit operation is similar to the view option: the user enters the desired last name, and once the last name is uniquely resolved, the corresponding IDP information is printed to the screen as a form. In the case of edit, however, the information fields within the IDP form are uniquely numbered from 1 to N, where N is a function of the number of training events in a given IDP. Minimal editing instructions are provided, but it is fairly intuitive and easy to comprehend (this editing scheme saw lots of use, since it was a primary element of the program used to construct the initial database).

Essentially, the user may edit any information field, within the IDP form, by entering the number that corresponds to it at the prompt. Data in the personnel information fields may be edited (i.e., to fix a spelling error or to change an entry); while, training events may be added, deleted or edited. After each edit operation, the form is reprinted to the screen to reflect the new information. Note that during an edit operation, the user may edit any number of fields, any number of times. After the user is finished editing the form, the tables in the run time environment are updated and the revised information is written into the two database files.

Once the delete operation has been selected, the user is requested to input the last name of the desired individual. Currently, the user must enter a last name (the thinking being that the user consciously selected the delete option). This approach may have been naive, and it will be revisited in the next revision. If the last name is unique to the database, the deletion takes place immediately. Otherwise, a list containing the last name, first name and middle initial of all IDP's with that last name is printed to the screen. Upon selection of the desired name, the deletion occurs. Deletion of information within the database system involves marking the entries as deleted in the run time tables; and rewriting the two database files to eliminate the information.

After selecting the add operation, the user is stepped through a series of prompt-response transactions to input the new IDP information. One nice feature is that the user is not required to be cognizant of the format internal to the database, since he user's response is reformatted to maintain consistency within the database. For example, if a user enters gs11 in response to the grade prompt, the program automatically converts it to GS11 since that is the format currently being used with the grade field. Upon completion of the input process, the user is placed into the edit operation; and, editing (as discussed above) may be performed on this new information. Once the user is satisfied that the new information is correct: the new IDP information is inserted into the run time tables; and, it is written into the two database files.

4. DISCUSSION

As mentioned in the introduction, the IDP process was broken down into three separate tasks. The IDP's had to be collected, the initial database had to be constructed and the database manager had to be in place. To aid the formatting task, a program that accepted keyboard input and transformed it into the appropriate records and fields was provided to the people responsible for constructing the initial database. By mid Oct 94, 252 FY 95 IDP's had been collected, the initial database was constructed and the prototype database manager was up and running. Note that the 252 IDP's translated into 252 records in the personnel information file; and, 1684 records in the training event file (or an average of about 6.5 training events/person). In November 1994, in response to customer feedback, the database manager was modified to improve to the user interface and to rework the form of the output generated by several of the search options.

Throughout this effort, the implementation of the prototype database system has been viewed as a work in progress. This philosophy is based on three observations. First, as the customer gains experience with the database system they will request changes. Second, when management reviews the IDP process at years end, changes may be required (i.e., based on this years experience, improvements may be implemented next year). And third, it is more efficient to optimize aspects of the system that are used, versus optimizing aspects of the system that "might" be used.

According to feedback from the primary customer, the database system is principally used for data retrieval. To date, at least two of the uses have the potential to be useful. The first involves identifying personnel who have indicated a desire to change specialty and then ensuring that they are on a path consistent with their own and the Command's goals. The second involves identifying personnel who have requested a specific course or development assignment. This information may be used to request that courses be brought on site, or to fill training slots that were allocated but can not be used as originally planned.

To acquire information about who accesses the database system (and how often), the database manager includes an audit routine. When the database system is invoked by a user, a line of text containing the user id, the date and the time is appended a log file. Figure 4 provides a visual break out of the users over the first three months of the database systems operation. As expected, personnel associated with the Professional Development Office are the primary users (86%). The amount of usage by the Department heads and others (e.g., Resource Control Office and Administrative Department) is small (about 8% and 6%, respectively). At least two possibilities exist; the Professional Development Office is the only outfit using the information; or, when others want the information they revert to the traditional approach of requesting it from the office that is responsible for knowing "that sort of thing" (this is not necessarily bad, but it would tend to confirm suspicions that tools alone will not increase productivity).

5. SUMMARY and FUTURE WORK

To assist the Engineering Directorate in its goal of improving the IDP process, a prototype database system was created. The database system was designed to meet the customers current needs and to be easily modifiable when new needs are identified (or, when old needs become obsolete). The database uses a relational data model and the database manager includes operations for data retrieval and data management. The database system has been in use since mid October 1994; and, in response to customer feedback, it has undergone one revision. An outline of future work follows.

5.1 Engineering Directorate Database Systems.

In addition to the IDP database system, there are several other stand alone databases being used within the Engineering Directorate. In general, each database is owned, maintained and used by different groups; they exist on different platforms (e.g., Personal Computers or CBDCOM Computers); and, they use different database management philosophies (e.g., spreadsheets, and commercial or custom database managers).

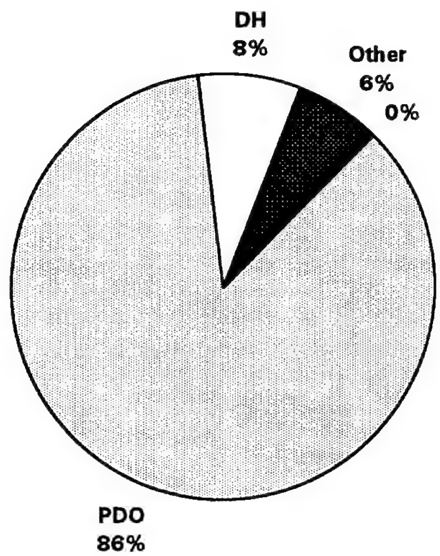

Figure 4. Break out of Engineering IDP database system users (PDO represents Professional Development Office personnel; and, DH represents the Department Heads).

Each database within the Engineering Directorate has taken on a responsible person, a name and a role. For example, someone in the Resource Control Office uses the "teams" database to track who is on what team for operational planning, someone in the Administrative Department uses the "admin" database to track civilian personnel actions, and someone in the Functional Area Office uses the "human resources" database to predict personnel requirements for strategic and long range planning. And now, someone in the Professional Development Office uses the "IDP" database to track IDP information. (It may be of sociological interest to note that the responsible persons name often serves double duty by substituting for the name of the database as well as the name of the person.)

Avoiding the organizational aspects of this, one real consequence is that the same kind of data (e.g., name, grade, department, etc.) is stored in various places; and, it doesn't always agree (i.e., the answer depends on who is asked). Note that varying data is not viewed as a problem by any given group, since each database is used for specific purposes; but, from a more global perspective, it can be disconcerting. As a result, several technical efforts are currently under way to more efficiently share the information within the Engineering Directorate.

5.2 Engineering Directorate IDP Process.

One way to further simplify the directorates IDP process, is to automate the IDP data collection task. The approach under consideration involves modifying the input program that was used this year (in the format the data task); and, making it accessible to all Engineering Directorate personnel via the CBDCOM computers. Modifications involve including more instructions; providing the user with their current personnel information (as contained in the IDP database), and altering the output routines. Current thinking is that the output would consist of a generated form mailed to the appropriate Department Heads (for review/approval); and, the formatted data mailed to the Professional Development Office (for inclusion in the database).

BLANK

LITERATURE CITED

[1] M. J. Rochkind, *Advanced UNIX Programming*, Prentice Hall Inc., 1985.
[2] C. J. Wertz, Relational Database Design: a practitioner's guide, CRC Press Inc., 1993.
[3] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd Ed., Prentice Hall Inc., 1988.
[4] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 1989.

BLANK

APPENDIX A

1. IDP Top Level Menu

Data Retrieval Options:
----
f - FIND Dif. (Pres/Fut Cat)
s - SEARCH the Data Base
v - VIEW an IDP entry

Data Management Options:
----
a - ADD an IDP entry
d - DELETE an IDP entry
e - EDIT an IDP entry

Navigation Command:
----
q - Exit Program

Input the appropriate letter:

2. Comments.

a. The current policy is that any user may retrieve information; but, only a small subset of the users is permitted to perform the management operations (i.e., write to the database).

b. The user selects a data retrieval option (Find differences, Search or View), or a data management option (Add, Delete or Edit), or exits the program by entering the appropriate letter at the prompt. Selection of a data retrieval option or a data management option produces a second level menu.

BLANK

APPENDIX B

1. **Sample IDP Form** (generated by the view form option).

VAX ID	LAST NAME		FIRST NAME	MI	GRADE
pxdrake	Drake		Paul	X.	GS12

SERIES Pres/Fut	ACTEDS Pres/Fut	SPECIALTY Pres/Fut	DEPT	TEAM LDR VAX ID
0893 / 0893	16 / 16	02 / 23	ENE	pemason

INDIVIDUAL TRAINING NEEDS (PRESENT)

COURSE TITLE	G/N	SOURCE	DATE
Mgt of Def Acq Contracts	G	ALMC	2Q95
Env Law	N	JHU	4Q95

INDIVIDUAL TRAINING NEEDS (FUTURE)

COURSE TITLE	G/N	SOURCE	DATE
Proj Plan & Ctrl Tech	G	ALMC	FY96
Solid Waste Treatment	N	JHU	FY96
Investigative Techniques	N	JHU	FY96
Ldrship & Team Bld	G	AMEC	FY97

INDIVIDUAL DEVELOPMENT ASSIGNMENT DESIRED? Y

TYPE	WHAT	WHERE	WHEN
OJT	Chem Engr	Chem Demil	FY96

2. **Interpretation.**

According to the personnel information on the above form, one can infer that Paul Drake is a Chemical Engineer (current series 0893); he is a member of the Scientists and Engineers Career Program (current ACTEDS code 16); he works in the area of systems engineering/development (current specialty code 02); and, for purposes of record keeping, he belongs to the Engineering Department (ENE) of the Engineering Directorate. In addition, because his future specialty code does not match his current specialty code, Mr. Drake indicates his desire to change his specialty area to program management, (specialty code 23).

With respect to training events: Mr. Drake has requested two training courses in FY 95; three training courses and a development assignment in FY 96; and, one training course in FY 97 (he probably views the specified course as a place holder). One can infer that in the short term (FY 95/96) he plans to increase his technical skills, primarily in the area of environmental engineering and, that in the longer term (after FY 96) he recognizes the need to increase his people skills (i.e., the request for leadership and team building training).